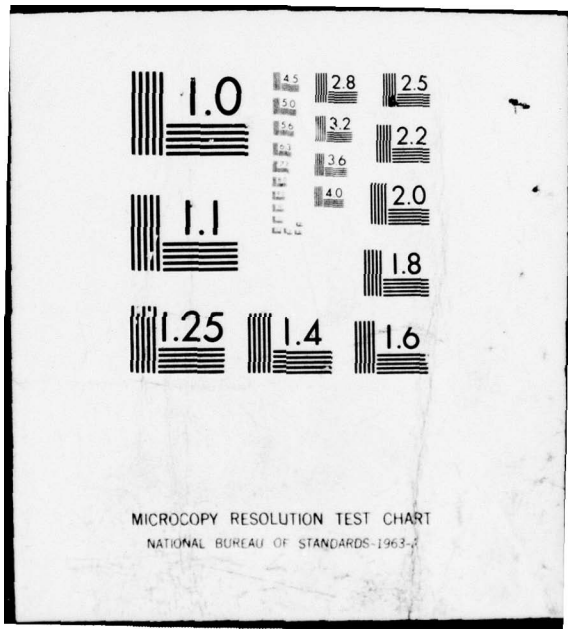
AD-A040 085 SYRACUSE UNIV N Y DEPT OF INDUSTRIAL ENGINEERING AND--ETC F/G 14/2
MANUAL FAULT DETECTION TEST SET MINIMIZATION.(U)
MAY 77 J KNAIZUK F30602-71-C-0312

UNCLASSIFIED TR-76-9 RADC-TR-77-149 NL

| OF |
AD
A040 085

END

DATE
FILMED
6-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

RADC-TR-77-149
Final Technical Report
May 1977

MANUAL FAULT DETECTION TEST SET MINIMIZATION

Syracuse University

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED: Anthony Coppola

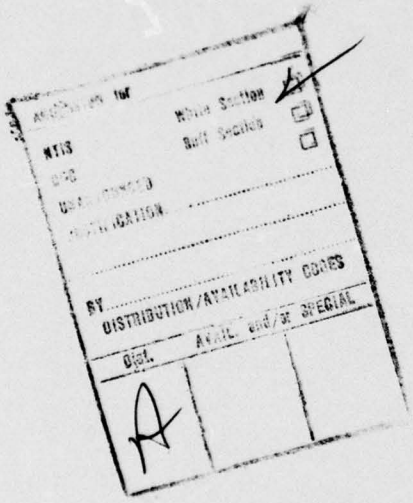
ANTHONY COPPOLA
Project Engineer

APPROVED: Joseph J. Naresky

JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office



Do not return this copy. Retain or destroy.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-149	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): MANUAL FAULT DETECTION TEST SET MINIMIZATION.	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. Jun 76 — Oct 76
6. PERFORMING ORG. REPORT NUMBER: 14 TR-76-9	
7. AUTHOR(s): J. Knaizuk, Jr.	8. CONTRACT OR GRANT NUMBER(s): 15 F30602-71-C-0312
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Syracuse University, Department of Industrial Engineering, Syracuse NY 13210	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 16 45400526, 17 05
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (RBRT), Griffiss AFB NY 13441	12. REPORT DATE: May 77
	13. NUMBER OF PAGES: 19 (12) 25p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Anthony Coppola (RBRT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reliability
LSI
Testing
Fault Detection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a manual procedure for minimizing the number of tests necessary to detect a single "stuck-at" fault in a large scale integrated circuit.

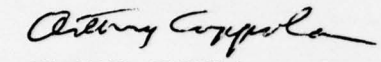
DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CDE 409 184

EVALUATION

This report presents a manual approach to minimizing the tests required for detecting faults in a Large Scale Integrated (LSI) circuit. It is not a "cookbook" procedure which can be directly applied to any device of interest, but is rather a heuristic approach which can be expanded and mechanized to fit the particular needs of those responsible for LSI testing. Since it addresses a pressing need for reducing LSI test complexity, it should be of great interest to microelectronic manufacturers and procuring agencies.



ANTHONY COPPOLA
Project Engineer

Manual Test Set Minimization

This work is the second part in a series on manual testing procedures.[1] A minimization procedure on a typical test set [2] will be analyzed by a step-by-step explanation of the reasoning involved. As in the first part of this series,[1] the test set generated to test the Arithmetic Logic Unit SN 54181 (See Figure 1) will be used in the minimization procedure. Let us first review how the test set was obtained.

There were 21 tests that were found to be required in order to fully test the device under test (DUT). By a method of logically dividing the DUT into five distinct partitions A, B, C,D, E (See Figure 2), a set of 8 common tests were found to test the four partitions A, B, C and D. Another 13 tests were required to test the final partition, E. (See Figure 3).

In the minimization procedure that follows the initial 8 tests will try to be merged into the final 13 tests to produce a required test set of only 13 tests. As will be seen, this combination process will be possible for only 7 of the 8 tests which will leave us with a total of 14 tests.

---

1 See RADC TR-75-17; Manual Testing Procedure For Detecting Single "Stuck-At" Faults, by John Knaizuk Jr. February 1975. AD#B002963L.

2 A test set in this case is a collection of tests required to find all detectable single "stuck-at" faults in the device under test.

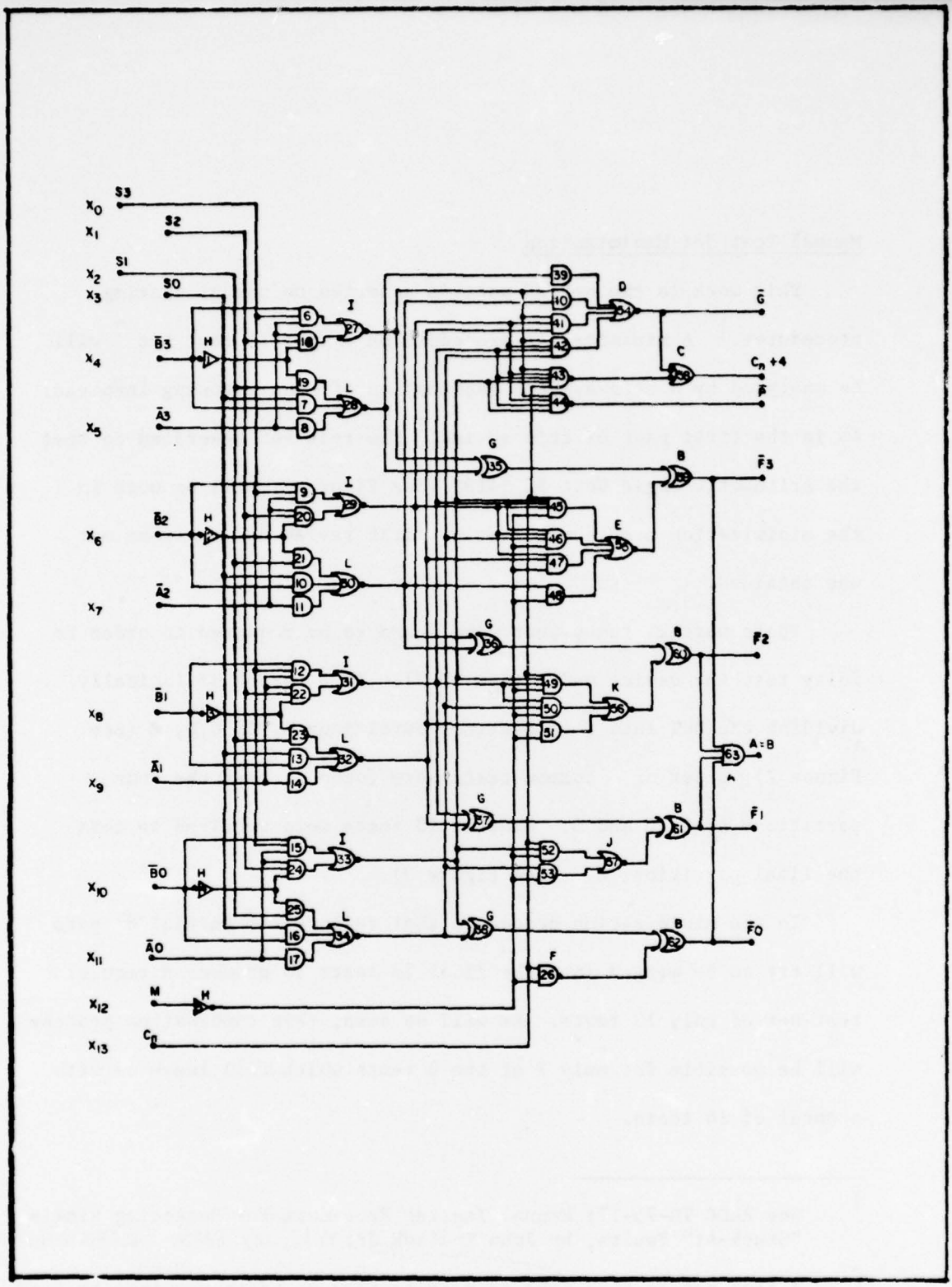


FIGURE 1. Logic Diagram for Arithmetic Logic Unit SN 54181

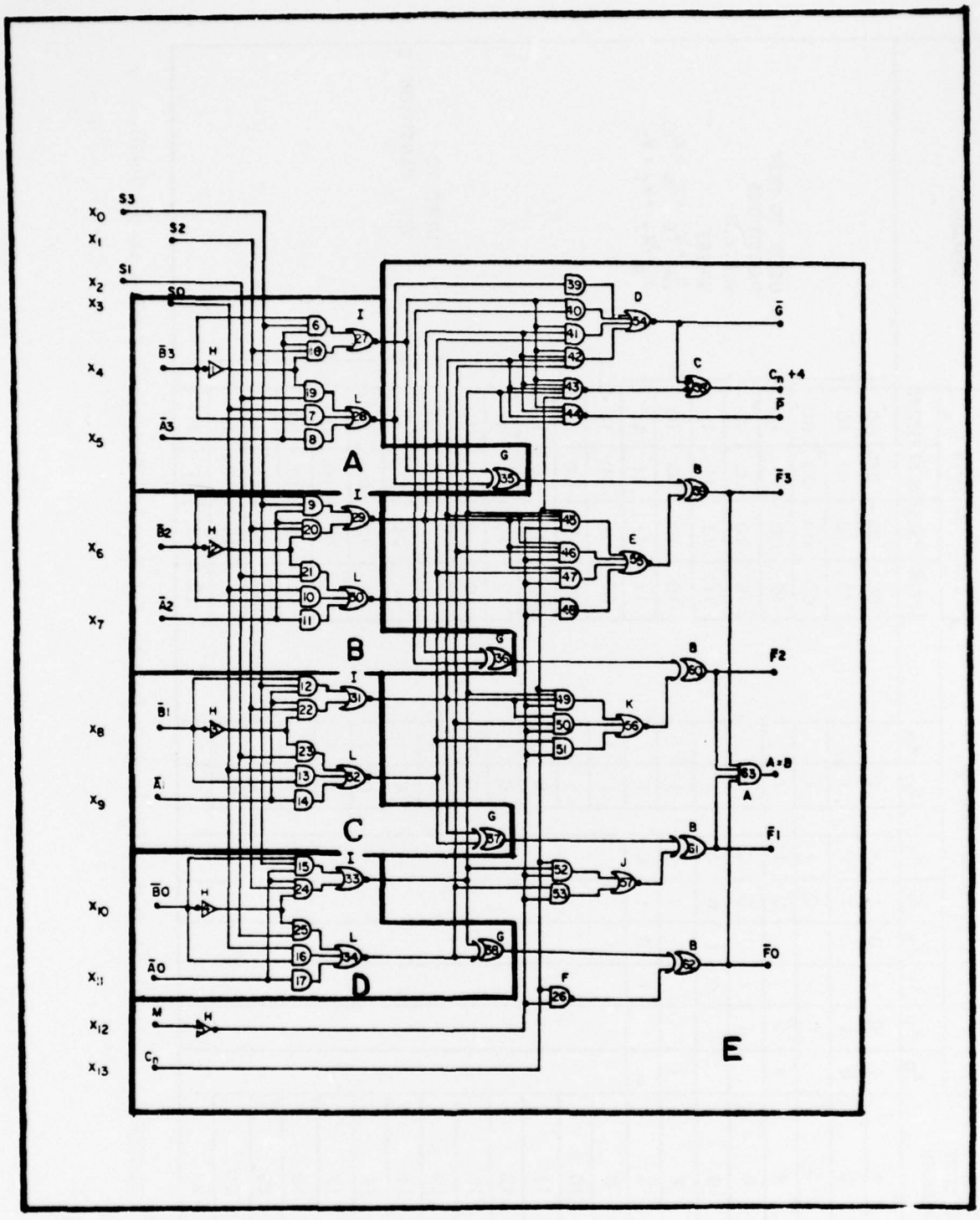


FIGURE 2. Partitioned Circuit SN 54181

TEST VECTOR NUMBER	PRIMARY INPUTS								GATE OUTPUTS				REMARKS
	X_0	X_1	X_2	X_3	X_4	X_5	X_{12}	X_{13}	(27,28)	(29,30)	(31,32)	(33,34)	
1	1	0		0	1	1	1		00	00	00	00	USED TO TEST PARTITIONS A,B,C,D. WHERE $X_4 = X_6 = X_8 = X_{10}$ $X_5 = X_7 = X_9 = X_{11}$
2	0	1			1	1	1		10	10	10	10	
3		1			0	1	1		00	00	00	00	
4	1	0			0	1	1		10	10	10	10	
5		1	1		0	0	1		10	10	10	10	
6			0	1	0	0	1		11	11	11	11	
7	1			1	1	0	1		10	10	10	10	
8			1	0	1	0	1		11	11	11	11	
9							1	1	00	11	11	11	USED TO TEST PARTITION E
10									10	11	00		
11							0	1	10	10	10	10	
12							0	1	10	10	10	11	
13							0	0	10	10	11	10	
14							0	1		11	00	00	
15							0	1	10	00	11	11	
16							0	1	10	10	00	11	
17							0	1		10	10	00	
18							0	0	10	10	10	10	
19							1		00	10	11	11	
20							1		11	11	10	11	
21							1		10	11	11	11	

FIGURE 3. Initial 21 Tests Required to Detect All Single "Stuck-at" Faults in SN 54181

Let us first observe that the primary input [3] x_{12} is in a logic 1 state for all of the first 8 tests. This was initially done so as to determine what the output would be at the primary output pins. We now remark that since this input (x_{12}) can only effect the monitored output values from gates 59, 60, 61, and 62 which are Exclusive - OR gates, the outputs of partitions A, B, C, and D will not be masked from these primary outputs. That is, the 8 tests for partition circuits A, B, C, and D are still valid no matter what value x_{12} possesses. An analogous argument holds for the primary input x_{13} even though it is not specified in the first 8 tests. With this in mind we may respecify the first 8 tests with "don't-cares" for inputs x_{12} and x_{13}.

The only problem which now exists is to match the outputs of the paired gates (27, 28), (29, 30), (31, 32), and (33, 34) of the (first) 8 tests on partitions A, B, C, and D with the required inputs from these same paired gates for the (last) 13 tests on partition E. Note that these paired gates are internal points within the DUT and as such are not true input/output connections but can be easily monitored by paths through Exclusive - OR gates to a primary output. Now even though these paired gates are shown having the same output for each

---

[3] A primary input (output) is a physical input (output) pin that test leads may be attached to.

test within the first 8 tests, our procedure will be to find a way of allowing us to mix the paired outputs. To this end we must examine the primary inputs x_0, x_1, x_2, and x_3 to find what values can be placed on these primary inputs with the possible combinations of (00), (10), and (11) [4] that can be placed on the paired gates (27, 28), (29, 30), (31, 32), and (33, 34). Table 1 gives a condensed look at the first 8 tests. Table 2 groups the tests with the same gate outputs. Table 3 shows all the probable combinations that can be made with the first 8 tests in order to obtain different simultaneous outputs from the paired gates (27, 28), (29, 30), (31, 32), and (33, 34).

Table 4 lists all probable test combinations given in Table 3 and indicates which are valid. Finally Table 5 summarizes the valid test combinations found in Table 4 and indicates the total number of times a test is used in combination with other tests.

We see that it is possible to obtain the desired combinations that we require. Now let us try to incorporate the first 8 tests into the remaining 13.

---

[4] The output of (01) from the paired gates is impossible to obtain on the DUT.

TEST NUMBER	PRIMARY INPUTS				PAIRED GATE OUTPUT
	x_0	x_1	x_2	x_3	(x_i, x_i+1)
1	1	0		0	(00)
2	0	1			(10)
3		1			(00)
4	1	0			(10)
5		1	1		(10)
6			0	1	(11)
7	1			1	(10)
8			1	0	(11)

where $x_i \in \{27, 29, 31, 33\}$

TABLE 1. Test with Required Inputs and Associated Logic States of Paired Gates.

NOTE: ALL BLANKS INDICATE "DON'T CARE" CONDITIONS.

PAIRED GATES OUTPUT (x_i, x_i+1)	TEST NUMBERS
(00)	1, 3
(10)	2, 4, 5, 7
(11)	6, 8

where $x_i \in \{27, 29, 31, 33\}$

TABLE 2. Output of Paired Gates at Specified Tests

PAIRED GATES OUTPUT $(x_i, x_i+1), (x_j, x_j+1)$	PROBABLE TEST COMBINATIONS
(00) (10)	(1,2) (1,4) (1,5) (1,7) (3,2) (3,4) (3,5) (3,7)
(00) (11)	(1,6) (1,8) (3,6) (3,8)
(10) (11)	(2,6) (2,8) (4,6) (4,8) (5,6) (5,8) (7,6) (7,8)
$(x_i, x_i+1), (x_j, x_j+1), (x_k, x_k+1)$	
(00) (10) (11)	(1,2,6) (1,2,8) (1,4,6) (1,4,8) (1,5,6) (1,5,8) (1,7,6) (1,7,8) (3,2,6) (3,2,8) (3,4,6) (3,4,8) (3,5,6) (3,5,8) (3,7,6) (3,7,8)

where $x_i, x_j, x_k \in \{27, 29, 31, 33\}$

and $x_i \neq x_j \neq x_k \neq x_i$

TABLE 3. Probable Test Combinations for the Desired Simultaneous Paired Gates Outputs.

COMBINED TESTS	VALIDITY OF COMBINATION
(1,2)	NOT VALID (N.V.) SINCE $x_0 = 1$ FOR TEST 1, $x_0 = 0$ FOR TEST 2
(1,4)	VALID
(1,5)	N.V. $x_1 = 0$ FOR TEST 1, $x_1 = 1$ FOR TEST 5
(1,7)	N.V. $x_3 = 0$ FOR TEST 1, $x_3 = 1$ FOR TEST 7
(3,2)	VALID
(3,4)	N.V. $x_1 = 1$ FOR TEST 3, $x_1 = 0$ FOR TEST 4
(3,5)	VALID
(3,7)	VALID
(1,6)	N.V. $x_3 = 0$ FOR TEST 1, $x_3 = 1$ FOR TEST 6
(1,8)	VALID
(3,6)	VALID
(3,8)	VALID
(2,6)	VALID
(2,8)	VALID
(4,6)	VALID
(4,8)	VALID
(5,6)	N.V. $x_2 = 1$ FOR TEST 5, $x_2 = 0$ FOR TEST 6
(5,8)	VALID
(7,6)	VALID
(7,8)	N.V. $x_3 = 1$ FOR TEST 7, $x_3 = 0$ FOR TEST 8
(1,2,6)	N.V. SEE (1,2) ABOVE
(1,2,8)	N.V. SEE (1,2) ABOVE

TABLE CONTINUED ON NEXT PAGE

COMBINED TESTS	VALIDITY OF COMBINATION
(1,4,6)	N.V. SEE (1,6) ABOVE
(1,4,8)	VALID
(1,5,6)	N.V. SEE (1,5) ABOVE
(1,5,8)	N.V. SEE (1,5) ABOVE
(1,7,6)	N.V. SEE (1,7) ABOVE
(1,7,8)	N.V. SEE (1,7) ABOVE
(3,2,6)	VALID
(3,2,8)	VALID
(3,4,6)	N.V. SEE (3,4) ABOVE
(3,4,8)	N.V. SEE (3,4) ABOVE
(3,5,6)	N.V. SEE (5,6) ABOVE
(3,5,8)	VALID
(3,7,6)	VALID
(3,7,8)	N.V. SEE (7,8) ABOVE

TABLE 4. Validity of Proposed Test Combinations.

PAIRED GATE OUTPUTS	TESTS USED 1	2	3	4	5	6	7	8
(00) (10)	X			X				
			X		X			
			X				X	
(00) (11)	X							X
			X			X		
			X					X
(10) (11)		X				X		
		X						X
				X		X		
				X				X
					X			X
						X	X	
(00) (10) (11)	X			X				X
		X	X			X		
		X	X					X
			X		X			X
			X			X	X	
TOTAL NUMBER OF TIMES EACH TEST WAS USED IN A COMBINATION.	3	4	8	4	3	6	3	8

TABLE 5. Valid Test Combinations to Produce Desired Logic Levels on Paired Gates (27, 28), (29, 30), (31, 32), and (33, 34).

First we shall see if it is possible to place the first 8 tests into the next 13 without regard to whether it is possible to implement these combinations. From Table 1 we see that for the first 8 tests, 2 tests have an output from their paired gates of (00), 4 tests have an output of (10), and 2 tests have an output of (11). Examining the required outputs of these paired gates in the last 13 tests we see that it is possible to match only 7 of the first 8 tests with their paired outputs. In the column for the paired gates (29, 30) of Table 1, the last 13 tests only have one required (00) (test 15) while 2 tests of the first 8 need a (00) output for this pair of gates (tests 1 and 3). Therefore it is impossible to incorporate one of these 2 tests into the last 13 tests.

We shall now assume that test 1 must be kept as is and try to merge the remaining 7 tests into the last 13 tests.

From Table 5 it is seen that tests 4, 5 and 7 have only one possible combination each to obtain the desired paired gates combination of (00) (10) (11). Further, tests 5 and 7 have the smallest number of times they can be used in different combinations of different paired outputs. See the last row in Table 5. Therefore, we shall incorporate these two tests first into the 13 tests for partition E. It is seen

from Figure 3 that tests 11 and 18 require (10) outputs for all paired gates, the same as tests 5 and 7. Hence the two tests 11 and 18 are equivalent to tests 5 and 7. This eliminates tests 11 and 18 from being chosen for any of the remaining 5 tests for partitions A, B, C, and D that are left.

Now we take tests 4 and 2 next because they have the least number of possible combinations available to produce various paired outputs. Since tests 2 and 4 both require (10) for their paired gates the tests 12 and 13 are the initial candidates for the merge. Note that when we choose a test as a candidate for replacing another test, we choose a test whose outputs match as closely as possible to the test we hope to replace. Since test 13 has a (11) for (31, 32) and we wish to complete (10) for test 1 or 4, a likely test to use for a (10) in (31, 32) is test 20. Test 20 also contains (11) for all other paired gates. We observe from Figure 3 and Table 5 that we can assign tests 4 and 6 to be replaced by tests 13 and 20. Note that test 4 was chosen before test 2 (which would have also worked in the above replacement) since it has less 3 way possible combinations for (00) (10) (11). The possible choices for the (11) outputs are tests 6 and 8, but test 6 has less ways it can be combined with other tests, so it was chosen above test 8.

Test 2 has the only required (10) outputs left but it seems that there are not any (10) outputs in column (33, 34) of the remaining tests left to fulfill this requirement. Test 10 is the answer to this problem. The blank in column (33, 34) in Figure 3 represents a "don't-care" situation. We shall assign the value (10) to (33, 34) in test 10. Now combining tests 10 and 12 we can fulfill the requirements for the replacement of test 2.

Since there is a (11) left from test 12, and parts of tests 2 and 8 can be specified at the same time (see Table 5), we shall use the (11) in column (33, 34) of test 12 for test 8.

The only tests left to be replaced are tests 3 and part of test 8. Both have the same number of ways to be combined with other tests. Which one should we choose to replace first? There are two important factors to consider here. First, column (27, 28) does not have a specified (11) paired output but does have a blank (don't-care") space (test 14) that can be used for this. The second factor to keep in mind is that there is only one place that (00) is specified in column (29, 30) (test 15) and this must be used for test 3.

Since there is no conflict between the required replacement tests, the choice will be to specify test 3 with test 15 first, then specify test 18 with the "don't-care" in column (27, 28). Now finish specifying

the rest of test 8 since it has at most two paired outputs not replaced by other tests while test 3 has 3 paired outputs not replaced.

The above argument for test 3 and test 8 was somewhat of an academic exercise since Table 5 indicates that these tests may be specified together. So let us proceed to replace (00) of test 3 with test 15 and (11) in both (31, 32) and (33, 34) of test 8 also with test 15. By specifying a (11) in (27, 28) of test 14 and using the (11) in (29, 30) of this test too, we complete the replacement of test 8. The (00) in (31, 32) and (33, 34) of test 14 can also be used for test 3. The (00) in column (27, 28) of test 9 is finally used to completely replace test 3. Figure 4 summarizes the above in a tabulated form.

Upon investigating test 21 we find that it is used just to check gate 59 for a stuck-at-1 fault condition. Therefore we may easily modify the paired outputs going into partition E while still keeping the test for gate 59 valid. A logical choice is to change all the (11) paired outputs to (00). See Figure 3, columns (29, 30), (31, 32), and (33, 34) of test 21. If we now replace the paired outputs (00) in column (27, 28) of test 1 with test 19 this will allow us to incorporate the rest of test 1 into test 21.

TEST VECTOR NUMBER	GATE OUTPUTS/MATCHING TESTS							
	(27,28)		(29,30)		(31,32)		(33,34)	
1	(00)	NO MATCH	(00)	NO MATCH	(00)	NO MATCH	(00)	NO MATCH
2	(10)	T12	(10)	T12	(10)	T12	(10)	T10
3	(00)	T 9	(00)	T15	(00)	T14	(00)	T14
4	(10)	T13	(10)	T13	(10)	T20	(10)	T13
5	(10)	T11	(10)	T11	(10)	T11	(10)	T11
6	(11)	T20	(11)	T20	(11)	T13	(11)	T20
7	(10)	T18	(10)	T18	(10)	T18	(10)	T18
8	(11)	T14	(11)	T14	(11)	T15	(11)	T15
9	(00)	T 3	(11)		(11)		(11)	
10	(10)		(11)		(00)		(10)	T 2
11	(10)	T 5	(10)	T 5	(10)	T 5	(10)	T 5
12	(10)	T 2	(10)	T 2	(10)	T 2	(11)	
13	(10)	T 4	(10)	T 4	(11)	T 6	(10)	T 4
14	(11)	T 8	(11)	T 8	(00)	T 3	(00)	T 3
15	(10)		(00)	T 3	(11)	T 8	(11)	T 8
16	(10)		(10)		(00)		(11)	
17	(11)		(10)		(10)		(00)	
18	(10)	T 7	(10)	T 7	(10)	T 7	(10)	T 7
19	(00)		(10)		(11)		(11)	
20	(11)	T 6	(11)	T 6	(10)	T 4	(11)	T 6
21	(10)		(11)		(11)		(11)	

FIGURE 4. Paired Gates and Their Specific Replacement Tests.

TEST VECTOR NUMBER	PRIMARY INPUTS															PRIMARY OUTPUTS							
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}		44	54	58	59	60	61	62	63
1A		1	0	1	0	1	0	0	0	0	0	0	1	1		1	1	0	1	1	1	1	1
2A	0	1	0	1	1	1	0	0	0	1	1	1	1			1	0	1	0	1	0	0	0
3A		1	1		0	0	0	0	0	0	0	0	0	1		0	1	1	1	1	1	1	1
4A	0	1	0	1	1	1	1	1	1	1	0	0	0	1		0	0	1	1	1	1	1	1
5A	1	0	0	1	0	1	0	1	0	0	0	1	0	0		0	0	1	1	1	1	0	0
6A		1	1	0	1	0	1	0	0	1	0	1	0	1		1	0	1	0	1	1	0	0
7A	0	1	1	0	1	1	0	1	1	0	1	0	0	1		1	1	0	0	0	0	0	0
8A	0	1	1	0	1	1	1	1	0	1	1	0	0	1		1	1	0	0	0	0	0	0
9A	1	0	1	0	1	0	0	1	0	1	1	1	0	1		0	0	1	1	1	0	0	0
10A	1			1	1	0	1	0	1	0	1	0	0	0		0	1	0	0	0	0	0	0
11A	1	0	1	0	1	1	0	1	1	0	1	0	1			1	1	0	1	0	1	1	0
12A	1	0	0	1	0	0	0	0	0	1	0	0	1			0	0	1	1	1	0	1	0
13A	1	0		0	0	1	1	1	1	1	1	1	1			1	0	1	0	1	1	1	0

FIGURE 5. Minimized Test Set

Figure 5 shows the final 13 tests required to fully test the DUT for all detectable "stuck-at" faults.

Conclusion

In summation, we see that the criteria for minimization of the test set was first to take a coarse overall look to see if there existed tests that could merge together, whether or not it could be implemented. If there were such a situation that some tests could not be incorporated, then we would exclude these tests from the minimization process. These excluded tests should be the ones that (1) require unobtainable logic gate levels and (2) have the smallest number of ways to combine with other tests that are also to be replaced (i.e. the number of possible combinations obtainable as noted in Table 5).

In the minimization process the main step (1) is to first replace the test that has the minimum number of ways it can be used in different possible combinations. This is done so that as less and less tests are left to merge with, the tests that still remain to be replaced will have more possible ways to be combined. Therefore, we have more of a change of incorporating these tests with others.

The next step that follows is (2) choose a replacement test that best satisfies the test to be replaced (e.g. if the test to be replaced has the paired outputs ((10), (10), (10), (10)), then the test chosen to replace it should have as many (10) paired outputs as possible).

Step (3): after partial replacement of a test use as much of the unassigned portions of the replacement test for other tests that still need to be merged.

Continue the above process until all tests have been replaced or you find that you can not merge what you have left.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.